

Please type a plus sign inside this box --> ☐

PTO/SB/21 (12/97)
Approved for use through 09/30/00 OMB 0651-0031

Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(To be used for all correspondence after initial mailing)

MAR 30 1998

Application Number	08/646,503
Filing Date	May 8, 1996
First Named Inventor	Guy Riddle
Group Art Unit	Unknown
Examiner Name	Not assigned
Attorney Docket No.	004860.P1937

TOTAL NUMBER OF PAGES IN THIS SUBMISSION 13

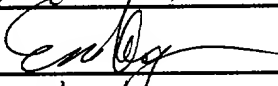
ENCLOSURES (check all that apply)

- | | | |
|--|--|--|
| <input checked="" type="checkbox"/> Fee Transmittal Form | <input checked="" type="checkbox"/> Assignment Papers | <input type="checkbox"/> After Allowance Communication Group |
| <input checked="" type="checkbox"/> Fee Attached | <input type="checkbox"/> Drawings | <input type="checkbox"/> Appeal Communication to Board of Appeals & Interferences |
| <input type="checkbox"/> Amendment/Response | <input type="checkbox"/> Licensing-related Papers | <input type="checkbox"/> Appeal Communication to Group (Appeal Notice, Brief, Reply Brief) |
| <input type="checkbox"/> After Final | <input type="checkbox"/> Petition Routing Slip (PTO/SB/69) & Accompanying Petition | <input type="checkbox"/> Proprietary Information |
| <input type="checkbox"/> Extension of Time Request | <input type="checkbox"/> Petition Checklist & Accompanying Petition | <input type="checkbox"/> Status Letter |
| <input type="checkbox"/> Express Abandonment Request | <input type="checkbox"/> To Convert a Provisional Application | <input checked="" type="checkbox"/> Additional Enclosure(s) (please identify below) |
| <input type="checkbox"/> Information Disclosure Statement | <input type="checkbox"/> Power of Attorney, Revocation, Change of Correspondence Address | <input checked="" type="checkbox"/> Declaration and Power of Attorney |
| <input type="checkbox"/> Certified Copy of Priority Document(s) | <input type="checkbox"/> Terminal Disclaimer | <input type="checkbox"/> Postcard |
| <input checked="" type="checkbox"/> Response to Missing Parts/Incomplete Application | <input type="checkbox"/> Small Entity Request | |
| <input type="checkbox"/> Response to Missing Parts Under 37 CFR 1.52 or 1.53 | <input type="checkbox"/> Request for Refund | |

Remarks

04/03/1998 ATHAII 00000039 08646503
01 FEB 105 130.00 DP

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

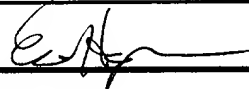
Firm or Individual Name	BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN Eric S. Hyman, Reg. No. 30,139
Signature	
Date	3/26/98

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on this date: March 27, 1998

Typed or Printed Name Eric S. Hyman

Signature



Date

3/26/98

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

HONORABLE COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231Sir:
Transmitted herewith for filing is the patent application of:Inventors: Guy RiddleFor: METHOD AND APPARATUS FOR DYNAMIC LAUNCHING OF A
TELECONFERENCING APPLICATION UPON RECEIPT OF A CALL

Enclosed are:

- XXX Eight (8) sheets of Formal/Informal Drawing(s) including eight (8) figures.
_____ An Assignment of the invention to: _____
_____ A Declaration and Power of Attorney.
_____ A Verified Statement to establish Small Entity Status under 37 CFR 1.9 and 37 CFR 1.27.

The Filing Fee has been calculated as shown below:

	(Col. 1)	(Col.2)	SMALL ENTITY		OTHER THAN A SMALL ENTITY	
For:	No. Filed	No. Extra	RATE	FEE	RATE	FEE
Basic Fee:	-	-	-	\$375.00	-	\$750.00
Total Claims:	9-20	-0-	x \$11.00		x \$22.00	-0-
Indep. Claims:	3-3	-0-	x \$39.00		x \$78.00	-0-
<input type="checkbox"/> Multiple Dep. Claim(s) Presented			+ \$125.00		+ \$250.00	-0-
* If the difference in (Col. 1) is less than zero, enter "0" in (Col. 2)			Total:	\$	Total:	\$750.00

XXX A check in the amount of \$750.00 to cover the filing fee is enclosed.
_____ A check for \$_____ covering Recordation of the Assignment is enclosed, along with the Assignment Cover Sheet.

XXX The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to our Deposit Account No. 02-2666. A duplicate copy of this sheet is enclosed.

XXX Any additional filing fees required under 37 CFR 1.16.

XXX Any patent application processing fees under 37 CFR 1.16.

XXX The Commissioner is hereby authorized to charge payment of the following fees during the pendency of this application or credit any overpayment to our Deposit Account No. 02-2666. A duplicate copy of this sheet is enclosed.

XXX Any extension or petition fees under 37 CFR 1.17.

XXX Any filing fees under 37 CFR 1.16 for presentation of extra claims.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMANDate: 5/8/96

Eric S. Hyman, Reg. No. 30,139

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025-
(310) 207-3800

CERTIFICATE OF MAILING:

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail No.

TB566363974US

in an envelope addressed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231 on

Nadya Gordon
Nadya Gordon

5/8/96
Date



Our Ref. No. 04860.P1937

UNITED STATES LETTERS PATENT APPLICATION

FOR

METHOD AND APPARATUS FOR DYNAMIC LAUNCHING OF A
TELECONFERENCING APPLICATION UPON RECEIPT OF A CALL

Inventor: Guy Riddle

Prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025

I hereby certify that this correspondence is being deposited with the
United States Postal Service as Express Mail No. TB566363974US
addressed to: Commissioner of Patents and Trademarks,
Washington, D.C. 20231 on May 8, 1996

Nadya Gordon
Nadya Gordon

5/8/96
Date



53,00 101 P

08/646503

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to the field of use of teleconferencing systems. More particularly, the present invention relates to the dynamic launching of teleconferencing applications upon receipt of a call.

Description of Related Art

10 Teleconferencing is increasingly becoming a popular application in personal computer systems. Such applications typically allow the transfer of audio and video data between users so that they can speak and otherwise communicate with one another. Such applications sometimes also include data sharing wherein various types of data such as documents, spreadsheets, graphic data, or other types of data, can be shared and manipulated by all participants in the teleconference. Different teleconference applications perhaps residing on different hardware platforms have different capabilities. Moreover, a wide variety of features has been implemented in different teleconference applications, and the proliferation of different types of computer systems with

15

20 different capacities, and different networking media has created challenges for teleconferencing.

For example, most systems which are used for teleconferencing applications are also used to run such programs for performing word processing, spreadsheet applications, database applications, and a variety of other applications. Thus, the

25

resources contained in the computer system are shared between these multiple applications. Often, most computer systems are limited in the amount of random access memory they contain and also the amount of processing power available for performing operations. In order for a user to receive calls, the user must keep the conferencing application open. Otherwise, if the called party does not have the conferencing application open, the calling party would receive a notice that the called party is not present to answer the call.

So, in order to receive a call, a user currently needs to keep any conferencing application open. However, keeping the conferencing application open is a waste of resources. Memory which is allocated to the conferencing application could be used for other applications. Also, processing resources are consumed in launching and maintaining the conferencing application. These resources are unnecessarily preoccupied for the times when there are no teleconferencing sessions in occurrence. A user can wait until he wishes to initiate a teleconferencing session before launching the teleconferencing application, but this means that there is no call notification unless the user receives a "regular" phone call, which does not allow for on-demand conferencing.

Thus, a solution needs to be provided that will allow a system to dynamically load the conferencing application only when necessary to answer a call, but not require the conferencing application to be loaded and executing to receive notice of an incoming call.

SUMMARY OF THE INVENTION

The invention provides a method and apparatus for dynamically launching a conferencing application upon the detection of an incoming call.

5 The invention can be implemented in a computer system having a memory, a processor, and a network interface by a method for dynamically launching a conferencing application upon the receipt of an incoming call having the steps of: receiving an incoming call signal on the network interface; processing the incoming call signal to detect an intended recipient application; and launching the intended recipient application.

10 An apparatus for dynamically launching a conferencing application upon the receipt of an incoming call is also shown, the apparatus having a call directing module; a process manager coupled to the call directing module; and, a conferencing component coupled to the network interface; and the call directing module; the conferencing component containing a circuit for notifying the call directing module upon receipt of an incoming call and causing the call director to signal the process manager to activate a conferencing application.

20 The invention will allow a system to dynamically load a conferencing application only when necessary to answer a call, but not require the conferencing application to be loaded and executing to receive notice of an incoming call.

Other objects, features and advantages of the invention will be apparent from the accompanying drawings, and from the detailed description that follows below.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates an example configuration in which various embodiments of the invention may be implemented.

Figure 2 illustrates a single system in which
5 embodiments of the invention may be implemented.

Figure 3 illustrates an example architecture on which a system employing various embodiments of the invention is based.

Figure 4 illustrates a preferences file configured in accordance to the invention.

10 **Figure 5** illustrates a system employing various embodiments of the invention.

Figure 6 illustrates a system employing various embodiments of the invention used for initializing persistent listening.

15 **Figure 7** illustrates a system employing various embodiments of the invention used for transferring an incoming call.

Figure 8 is a flow diagram illustrating a preferred operation of the invention used for dynamically launching a
20 conferencing application.

DETAILED DESCRIPTION OF THE INVENTION

The present invention provides a method and apparatus for dynamically launching teleconferencing applications upon receipt of a call. For purposes of explanation, specific
5 embodiments are set forth to provide a thorough understanding of the present invention. However, it will be understood by one skilled in the art, from reading this disclosure, that the invention may be practiced without these details. Further, although the present invention is described through the use of
10 certain specific embodiments thereof, especially, with relation to certain hardware configurations, data structures, packets, method steps, and other specific details, these should not be viewed as limiting the present invention. Various modifications can be made by one skilled in the art, without departing from the overall
15 spirit and scope of the present invention.

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent disclosure, as it appears in the Patent and Trademark
20 Office patent files or records, but otherwise reserves all copyright rights whatsoever. Copyright Apple Computer, Inc.

A typical system configuration in which a teleconference may take place is illustrated as 100 in **Figure 1**. For example, a first workstation 150 may communicate via teleconference with a
25 second workstation 155, as illustrated. System 150 may include a central processing unit 150c which is coupled to a display 150d a

video input device 150a, and a sound input device 150b. The system 150 may communicate with over system 155 over networking medium 170 via network connection module 160. Network connection module 160 may include any number of network adapters commercially available such as Ethernet, Token Ring, or any other networking standard commercially available. Note that network adapter 160 may also include a wireless network adapter which allows transmission of data between components without a medium 170. Communication is thus provided via network adapter 165 coupled to system 155, and bi-directional communications may be established between two systems. System 150 further has a keyboard 150e and a pointing device 150f, such as a mouse, track ball, or other device for allowing user selections and user input.

In implemented embodiments of the present invention, a general purpose computer system is used for implementing the teleconferencing applications and associated processes to be described here. Although certain of the concepts to be described here will be discussed with reference to teleconferencing, it is apparent that the methods and associated apparatus can be implemented for other applications, such as file sharing, real time data acquisition, or other types of applications which sends data from a first participant to a second participant or set of participants. A computer system such as that used for implementing embodiments of the present invention will now be described.

Figure 2 is a diagram showing a computer system capable of implementing the present invention, such as a workstation,

personal computer or other processing apparatus. The sub-system 300 comprises a bus or other communication means 301 for communicating information, and a processor 302 coupled with bus 301 for processing information. Sub-system 300 further comprises
5 a random access memory (RAM) or other volatile storage device 304 (referred to as main memory), coupled to bus 301 for storing information and instructions to be executed by processor 302. Main memory 304 also may be used for storing temporary variables or other intermediate information during execution of instructions
10 by processor 302. Sub-system 300 also comprises a read only memory (ROM) and/or other static storage device 306 coupled to bus 301 for storing static information and instructions for processor 302, and a mass storage device 307 such as a magnetic disk or optical disk and its corresponding disk drive. Mass storage
15 device 307 is coupled to bus 301 for storing information and instructions.

Sub-system 300 may further be coupled to a display 321 such as a cathode ray tube (CRT) or liquid crystal display (LCD) coupled to bus 301 for displaying information to a computer user.
20 Such a display 321 may further be coupled to bus 301 for the receipt of video or image information. A keyboard 322, including alphanumeric and other keys may also be coupled to bus 301 for communicating information and command selections to processor 302. An additional user input device is cursor control 323, such as a
25 mouse, a trackball, stylus, or cursor direction keys, coupled to bus 301 for communicating direction information and command selections to processor 302, and for controlling cursor movement

on display 321. For teleconferencing applications, system 300 may also have coupled to it a sound output device 328, a video input device 329, and sound input device 326, along with the associated D/A (Digital-to-Analog) and A/D (Analog-to-Digital) converters or software codecs for inputting or outputting media signal bitstreams. System 150c may further be coupled to communication device 327 which is coupled to network adapter 160 for communicating with other computers over network 370.

Note, also, that any or all of the components of system 150c and associated hardware may be used in various embodiments, however, it can be appreciated that any configuration of the system may be used for various purposes according to the particular implementation.

In one embodiment, system 300 is one of the Apple Computer® brand family of personal computers such as the Macintosh 8100 brand personal computer manufactured by Apple Computer, Inc. of Cupertino, California. Processor 302 may be one of the PowerPC brand microprocessors manufactured by Motorola, Inc. of Schaumburg, Illinois.

Although a general purpose computer system has been described, it can be appreciated by one skilled in the art, however, that the following methods and apparatus may be implemented in special purpose hardware devices, such as discrete logic devices, large scale integrated circuits (LSI's), application-specific integrated circuits (ASIC's), or other specialized hardware. The description here has equal application to apparatus having similar function.

Figure 3 illustrates a plurality of processes and/or apparatus which may be operative within system 150c. At the highest level, for example, at the highest level in the ISO/OSI networking model, a conferencing application 401, such as a teleconferencing application, an audio/video server, or a data server, communicates with a conference component 400 in the form of Application Program Interface (API) calls.

Conference component 400 allows conferencing application 401 to establish communications between two or more teleconference stations. Control information, and media information can be transmitted between the first participant system and a second participant system. Conference component 400 communicates with the transport component 402 by sending messages for other teleconferencing stations which are encapsulated and placed into a form that the transport component 402, and the network component 403, can packetize and transmit over networking medium 170.

Transport component 402 and networking component 403 provide the necessary operations for communication over the particular type of network adapter 160 and networking medium 170 according to a particular implementation. For example, networking component 403 may provide the TCP or ADSP protocols and packetizing, according to those respective standards. Transport component 402 can support standards such as H.320 or MovieTalk™ transport standards. There can exist multiple transport components and multiple network components, as described below.

The main function of conference component 400 is to establish and maintain a bi-directional connection between every

member of a conference -- i.e., between conferencing applications. Conferencing applications use a control channel to exchange control data that is pertinent to the conference. This data might include user identification information or other information that is germane to the application's operation. Conferencing applications (e.g., conferencing applications 401) define the format and content of these control messages by establishing their own control protocols within the boundaries of the conferencing API. Conferencing components further establish communication channels between a first endpoint and a second endpoint, using underlying transport component 402. Thus, once a media channel has been established, conference component 400 uses the media channel of transport component 402 which is provided for transmission of media and non-media information.

Conferencing application 401 controls conference component 400 by the issuance of QuickTime™ Conferencing API calls. Conferencing applications operate using an event-driven model wherein events pertinent to the application are issued to conferencing application 401. Conferencing application 401 can then take appropriate action either by modifying internal data structures within (creating a source or sync), and/or issuance of appropriate messages over the network to other connected components, or potential participants. In addition, conference components also respond to events and messages that are received. In addition, conference components take appropriate actions pertaining to the receipt of API calls from conferencing applications.

There can exist multiple conferencing components, wherein each conferencing application requires at least one conference component, but each conferencing application can have more than one associated conference component. Each conferencing component has an unique identification number. In addition, each conference component contains one "listen string", which is unique. A listen string is the encapsulation of the parameters of the "MTConferenceListen" API call for each conference component. Listen strings can contain more than one network or port. A listen string is composed of two parts: a fixed portion identifying a service name (which is similar to service names given to printers in an AppleTalk™ network that are displayed in the Chooser application in the Apple Macintosh operating system), and a variable portion containing a list of one or more service types, which contain the transport/network types with which the transport components and network components can interface. For example, service types can be port numbers for TCP/IP networks or device types for AppleTalk network. The transport/network tuples will be described below in association with the discussion of

Figure 5.

The system as shown in **Figure 3** requires that a conferencing application 401 be present to handle incoming call events generated by conference component 400. As conferencing applications (such as conferencing application 401) utilize significant system resources (e.g., processor processing power and memory space), the requirement that conferencing application 401 be executing even when there are no calls present to necessitate

the existence of a conferencing application prevents the use of those resources by other applications. A system that removes the requirement by allowing conferencing application 401 to be launched when needed (i.e., launching only when there is an incoming call to handle), is described below.

Figure 5 illustrates a preferred embodiment of the invention having a call director 502; a demon conference component 504 (i.e., a conference component acting in demon mode); a transport component 506; and a network component 508. The preferred embodiment also contains a call director preferences 510. Call director 502, demon conference component 504, transport component 506, and network component 508 can be referred to as call processing module.

Call director 502 is a "faceless" background process that is loaded at initialization of the computer system contained in **Figure 2**. One of the main functions of call director 502 is to initiate the automatic launching of a conferencing application when a call is received by the computer system. In addition, call director 502 is responsible for initiating and interacting with demon conference component 504 to control the transfer of calls to a conferencing application. As a faceless process -- i.e., a process that does not need to contain any code to interface directly with a user -- call director 502 requires very little in terms of system resources. More importantly, aside from the indications given by the dynamic launching capabilities and other functionality provided by call director 502, and the relatively small memory foot-print of call director 502, the user does not

even have to be aware that call director 502 is existent. Through the use of the elements contained in **Figure 5**, conferencing application 401 does not have to be loaded and executing until an incoming call exists.

5 Demon conference component 504, which is controlled by call director 502 through the use of the QuickTime™ Conferencing Application API, is responsible for performing the "persistent listening" for incoming calls. Demon conference component 504 is created by call director 504 after call director 504 has finished
10 launching. Demon conference component 504 is an instance of the class of conference components that is initiated into a special mode of operation by call direction 504 through the use of a "MTConferenceSetPersistence" API call with the parameter "mtPersistenceDemonMode".

15 In a preferred embodiment, there can only be one demon conference component in each computer system. Demon conference component 504 is the only conference component instance of call director 502. That is, call director 502 can only have a single instance of a conference component (as opposed to conferencing
20 application, which can have multiple conference component instances). Demon conference component 504 communicates with other conference components to transfer incoming calls indicated by transport component 506 and network component 508 using a shared data structure in memory. A preferred embodiment of the
25 shared data structure is further described below, along with a description of the basic operations of the invention, while referencing **Figure 6**.

Figure 6 illustrates a sample configuration using the preferred embodiment of the invention wherein conferencing application 401 and conference component 400 interacts with call director 502 and demon conference component 504 through the use of a shared queue structure 512.

Inter-Conference Component Communication

In the preferred embodiment, conference components communicate (i.e., achieve interprocess communication) through the use of shared memory. Specifically, conference components communicate through the use of globally accessible data structures composed of a demon queue and an application queue, both of which are contained in shared queue structure 512. The demon queue is used by any conferencing component of a conferencing application to send commands and information to demon conferencing component 504 ("QdPersistenceOn", "QdPersistenceOff", "QdListenAgain", "QdPersistenceClear"). The application queue is used by the demon conferencing component to send messages to other conferencing components ("QdListenerStatus", "QdDemonOff", "QdIncomingCall"). It is to be noted that the choice of using queues to allow inter-component communication is not intended to be limiting, and other methods of allowing inter-component communication can be used to achieve the same functionality. For example, instead of using queues to transfer commands and information, messages can be passed from one conferencing component to another. Alternatively, registers may be used to pass information from one conference component to another.

In the following description of **Figure 6**, it is assumed that call director 502 has been loaded at the time of initialization of the computer system, and call director 502 has created an instance of the class of conference components and initialized into that conference component instance into demon conference component 504 through the use of the "MTConferenceSetPersistence" API call with a parameter of "mtPersistenceDemonMode". It is important that a demon conference component such as demon conference component 504 exists so as to perform persistent listening. If there is not a conference component in demon mode, there can be no persistent listening. Moreover, if a conferencing application tries to turn on persistent listening when there is no demon conference component initiated, the conference component of the conferencing application will return a "mtDemonKaputErr" message, indicating that there is no demon conference component to turn-on persistent listening.

Setting-Up Persistent Listening

As stated above, demon conference component 504 is responsible for listening for incoming calls on behalf of all conferencing applications that request persistent listening. Call director 502 is responsible for dynamically launching (if necessary) and transferring an incoming call to the conferencing application which requested persistent listening. The process for configuring demon conference component 504 and call director 502 in the preferred embodiment is as follows:

(1) conferencing application 401 will first send an "MTConferenceSetPersistence" API command with an "mtPersistenceOnMode" parameter after being launched to conference component 400;

5 (2) conference application 401 will then send an API command ("MTConferenceListen") requesting persistent listening and passing a listen string, which includes the identification of the port on which it wishes demon conference component 504 to listen, to conferencing component 400;

10 (3) conference component 400 will place a request (QdPersistenceOn) on the demon queue to have demon conference component 504 perform persistent listening on the port specified by conferencing application 401 (the request containing an application signature, as discussed below, identifying
15 conferencing application 401 as the requester and the parameters, or so-called "listen string", of the listening that conferencing application 401 is requesting, the parameters including a service name and a port);

(4) demon conference component 504 will initialize transport
20 component 506 and network component 508 as necessary to perform persistent listening on the requested service type and port;

(5) at substantially the same time as step (4), demon conference component 504 will also notify call director 502 through the use of a "mtPersistenceChangedEvent" that conferencing
25 application 401 has requested persistent listening, and send the application signature of conferencing application 401 and the listen string, which, as stated, includes information regarding

the service type and port on which conferencing application 401 wishes to listen;

(6) call director 502 will then store the information received from demon conference component 504, including the application signature of conferencing application 401 (call director 502 will create an alias, as described below, for conferencing application 401 from the application signature), the service name, the transport type, the network type, and the service type into call director preferences 510; and,

(7) lastly, conferencing application 401 can either end execution or remain running -- but under either case, the listening for incoming calls will be done by demon conference component 504, as described below.

Persistent Listening of Incoming Calls

During normal operations, demon conference component 504, after detecting an incoming call, will notify the conferencing application which requested the listening to transfer the incoming call. As mentioned above, in order to ensure that an incoming call can be matched-up with a conferencing application, call director 502 uses call director preferences 510 to track of the conferencing applications that requests persistent listening. Call director 502 also uses call director preferences 510 to track all listen strings of the various conference components corresponding to the various conferencing applications. Also as discussed above, each listen string corresponds to a particular conference component and contain the service and the ports for

which that conference component is responsible. Thus, call director preferences 510 contains: (1) a list of aliases for conferencing applications that requested listening; and (2) what each conferencing applications want to listen on, such as the name of a user, the transport and the network type, and the service type (e.g., a port number for TCP/IP)).

Figure 4 illustrates the contents of call director preferences 510, displayed in content window 410, containing logical representations of: a listen strings list 412 ("mtls"), and a conferencing application alias list 414 ("alis"). Call director 502 uses call director preferences 510 to keep track of the persistent listening requests of conferencing applications, and to hold the values used to initiate a demon conference components (e.g., demon conference component 504), any transport components (e.g., transport component 506), and any network components (e.g., network component 508). The contents of listen strings list 412 is displayed in a listen string list window 416. The contents of conferencing application alias list 414 is displayed in a alias list window 418.

As can be seen in listen string list window 416, only one listen string, a listen string 420, is contained in listen strings list 412. Listen string 420 is identified in listen string list 412 by the unique identification number "20556", which is the identification number used to identify related resources in call director preferences 510. In addition, in listen string list window 416, it is shown that listen string 420 was initialized by conferencing application 401, which in this example is entitled

"QuickTime™ Web Conference". Thus, listen string 420 identifies that conference component 400 belongs to conferencing application 401.

The contents of listen string 420 is displayed in a listen string content window 422. Listen string 420 contains a service name 424 ("James Watt" in ASCII and a hexadecimal equivalent), a transport type 426 ("mtlktcpi" in ASCII and a hexadecimal equivalent), and a port 428 ("458" in ASCII and a hexadecimal equivalent). Thus, conferencing component 401 is the requester of persistent listening for transport type 426 and port 428.

Referring still to **Figure 4**, a conferencing application alias 430 is shown in conferencing application alias list 414 in alias list window 418. Conferencing application alias 422 has an identification number 20556, which is the same identification number used to identify listen string 420 in call director preferences 510. Conferencing application alias 422 is used by call director 502 to locate and launch conferencing application 401 (i.e., QuickTime™ Web Conference) when an incoming call matches the profile contained in listen string 420. The aliases contained in conferencing application alias list 414 is kept in call director preferences 510 and only used by call director 502 -- i.e. aliases are never passed down to demon conference component 504.

The contents of conferencing application alias 430 is shown in alias content window 432 and contains the location of conferencing application 401.

Answering of Incoming Calls After Persistent Listening Has Been Activated.

After persistent listening has been set-up, assuming that conferencing application is still running (see **Figure 6**),
5 when an incoming call is detected by transport component 506, demon conference component 504 will transfer the incoming call to conferencing component 400, which will notify conferencing application 401 of the incoming call. The incoming call is transferred through the following sequence:

- 10 (1) demon conference component 504 sends a "QdIncomingCall" message to conference component 400 through the use of shared queue structure 512;
- (2) conference component 400 creates a new instance of a transport component and a new instance of a network component,
15 which in **Figure 7** is transport component 402 and network component 403, respectively;
- (3) demon conference component 504 sends conference component 400 a reference to transport component 506;
- (4) conference component 400 "answers" the call by sending a
20 "MTTransportAnswer" message, along with the reference to transport component 506, to transport component 402 instance to transfer the call from transport component 506;
- (5) after the call has been transferred successfully, conference component 400 sends a "QdListenAgain" message to demon
25 conference component 504 through the use of shared queue structure 512; and,

(6) demon conference component 504 issues a "MTTransportListen" API call to transport component 506 to await the next incoming call.

5 System Re-Initialization After Persistent Listening has been
Initialized.

When the computer system is re-initialized and call director 502 is loaded and begins execution after system initialization, the following start-up sequence occurs:

- 10 (1) call director 502 reads call director preferences 510 and retrieves any listen strings;
- (2) call director 502 initializes demon conference component 504 to place it into demon mode as described above;
- 15 (3) call director 502 sends one "MTConferenceDemonListen" API call to demon conference component 504 for each listen string that is retrieved from call director preferences 510, where each API call passes demon conference component 504 the retrieved listen string and the associated application signature for the conferencing application that requested the listening.

Hi-jacking of listening

20 A later conferencing application will replace the listening of conferencing application 401 if the later conferencing application wants to listen to the same port (under TCP/IP) or the same name/device (under AppleTalk). If this occurs, a "mtListenHijackedErr", generated by demon conference component 504, will be received by conference component 400 if

25 conferencing application 401, which has been "hi-jacked," is still

running. Conference component 400 will then inform conferencing application 401 that the listening requested by conference component 401 has been taken over so that conferencing application 401 can take any necessary action.

5 In addition, demon conference component 504 will send a "MTConferenceSetPersistence" API call with the parameter of "mtPersistenceOffMode", along with the application signature of conferencing application 401, to call director 502. Call director 502 will then remove the listen strings for conferencing application 401 from call director preferences 510.

If conferencing application 401 is not running when a hi-jack occurs, then the "mtListenHijackedErr" will be removed after a certain time.

Turning Off Persistent Listening

15 If persistent listening is turned off for a listen string (i.e., a conference component), there will be no notification of incoming calls for that listen string if the conferencing applications that handles that listen string is not loaded and executing -- i.e., the system will operate as it had before the existence of the invention. However, the user will continue to receive notification of incoming calls on the listen strings for which persistent listening has not been turned off.

20 The sequence to turn off persistent listening will depend on whether conferencing application 401 is loaded and executing. If conferencing application 401 is loaded and
25 executing, then the sequence is as follows:

(1) conferencing application 401 sends conference component 400 a request to turn off persistent listening via a "MTConferenceSetPersistence" API call with "mtPersistenceOffMode" parameter;

5 (2) conference component 400 sends a "QdPersistenceOff" message to demon conference component 504;

(3) demon conference component 504 will then remove transport component 506 and network component 508 and send a "QdDemonOff" message to conference component 400;

10 (4) demon conference component 504 sends a "MTPersistenceChangedEvent" message to call director 502 with the application signature for conferencing application 401;

(5) call director 502 removes the listen string for conference component 400 from call director preference 510;

15 (6) conference component 400, after receiving the "QdPersistenceOff" message from demon conference component 504, will create a new instance of a transport component and a new instance of a network component and initialize them for local listening -- i.e. conference component 400 will be responsible for
20 waiting for an incoming call for the listen string.

If the user thereafter quits conferencing application 401, then the system will operate as if call director 502 is not present and the user will receive no notifications of incoming calls as conferencing application is not loaded and executed to
25 perform listening.

It is to be noted that as a listen string can have more than one transport component and network component created for

persistent listening -- e.g., a listen string contains the listening for both a TCP/IP port and a AppleTalk service -- demon conference component 504 will have to remove all the transport components and network components associated with the listen string for which persistent listening is turned off in step (3). In addition, when those instances of transport components and network components are removed, the conference component which requests that persistence listening be turned off for its listen string (e.g., conference component 400) will have to create a new set of transport component and network component instances to continue listening in step (6).

For a user to turn off persistent listening for the services and port that conferencing application 401 processes if conferencing application 401 is not currently loaded and executing, the user has to first launch conferencing application 401. Conferencing application 401 then reads its own preference files and performs listen with same values as it did the last time it executed (i.e., conference component 400 sends a listen request with the same listen string it sent to initiate persistent listening to demon conference component 504). Then, the same sequence used to turn off persistent listening is used, as described above.

Dynamic Launching of a Conferencing Application

Figure 8 is a flow diagram of the preferred operation of the invention wherein call director 502 operates to dynamically launch a conferencing application after persistent listening has

been initialized and an incoming call is received. The system in operation at the start of the flow diagram is as shown in **Figure 5**.

In block 802, call director 502 detects an incoming call through the use of demon conference component 504 and transport component 506. Call director 502 is notified by an "mtIncomingCallForEvent", containing an application signature of the conferencing application which set-up the listen string.

In block 804, demon conferencing component 504 will place a "QdIncomingCall" message on the application queue with the application signature, listen string, identity of transport component 506 (i.e., a reference to transport component 506), and an "MTAddress" parameter, which identifies the address of the caller. Demon conference component 504 will also send an "MTIncomingCallForEvent" to call director 502 that an incoming call has been received along with an application signature and listen string for conferencing application 401 and conference component 400. Call director 502 then checks the current process list to see if the conferencing application with the target application signature (i.e., conferencing application 401) is a process that is currently running. Operation will then continue with block 806, as discussed below. If the conferencing application is not running, call director 502 will try to launch the conferencing application, as discussed in block 808.

In block 808, where the conferencing application is not currently executing, call director 502 will determine if the conferencing application is locatable so that it can be launched

-- i.e., whether the location of the conferencing application can be ascertained. Call director 502 will retrieve conferencing application alias 422 for conferencing application 401 from call director preferences 510, update the location of conferencing application 401 if necessary, and then use the process manager to launch conferencing application 401. If a conferencing application corresponding to conferencing application alias 422 cannot be found (e.g., where conferencing application 401 has been removed from the storage devices accessible to the computer system), then operations will continue with block 824.

In block 810, if conferencing application is locatable, call director 502 will determine whether there is enough free memory to run the conferencing application. If there is enough memory for conferencing application 401 to execute, call director 502 will then initiate the launching of conferencing application 401 continuing with block 812.

If there does not exist enough memory for the conferencing application to execute, operations will continue with block 816, where the user will be notified through an alert dialog that conferencing application 401 does not have enough memory to launch, and unless the user terminates and quits one or more processes that are currently occupying memory, the user will not be able to accept the incoming call. Call director 502 will keep checking for the user to free up memory until a predetermined time-out period has elapsed in block 818. At the end of the time-out period, if the user has not freed-up enough memory, operation

will continue with block 826. If the user does free up enough memory, the operations will continue with block 812.

In block 812, where there exists enough memory for conferencing application 401 to begin execution, call director 502 will launch conferencing application 401 by using the process manager. Conferencing application 401 is notified that it must process the incoming call and therefore launches.

After conferencing application 401 has launched, the system configuration will be as shown in **Figure 6**, where conferencing application 401 and its associated conference component 400 has loaded and is executing.

In block 814, call director 502 checks to see if conferencing application 401 is listening in the same way as it was when the conferencing application set-up call director 502 for persistent listening. If conferencing application 401 does not listen in the same way within a reasonable time, demon conference component 504 recognizes that the incoming call has not been handled (i.e., the incoming call event has not been removed from the application queue) and will inform call director 502 with a "mtPersistenceChangedEvent" with the "mtPersistenceOffMode" parameter and the application signature of conferencing application 401. Call director 502 will then remove the entry for conferencing application 401 from call director preference file 510 in block 824, as described below. If conferencing application 401 is listening in the same way, then conferencing application 401 is transferred the incoming call as in block 806.

In block 806, and referring to **Figure 7**, after the conferencing application has completed launching, or if the conferencing application is already executed, call director 502 will transfer the incoming call to the conferencing application, as described above, and return to listening, as discussed in block 822, below.

After conferencing application 401 has been transferred the call, conferencing application 401 will then be responsible for giving the user an option to accept the call. If the user decides to accept the call, then conferencing application 401 will perform as usual and process the incoming call. If the user does not accept the call, then operation will continue with block 820. It is to be noted that whether or not the user decides to accept the call, call director 502 is not affected after call director 502 has transferred the incoming call to conferencing application 401 and returns to listening, as discussed in block 822.

In block 822, after either: (1) call director 502 has transferred the incoming call to the conferencing application as in block 806; or (2) demon conference component 504 has dropped the call -- i.e. removed the call from the incoming call event queue -- as in block 826, demon conference component 504 will return to listening.

In block 824, where conferencing application 401 is not locatable or conferencing application 401 is not listening using the same values with which conferencing application 401 set-up call director 502, call director 502 will remove all references to conferencing application 401 from call director preferences 510.

In block 826, if there is not enough memory available to launch conferencing application 401 and the user does not free-up any memory within the time-out period in block 816, then the incoming call will not be answered and the caller will receive a notice that the user the caller is trying to contact is not available. The incoming call will also be dropped if conferencing application 401 is not listening in the same way as it was when conferencing application 401 set-up call director 502 to listen for incoming calls. If there is not enough memory available to launch conferencing application 401 and the user does not free-up any memory within the time-out period in block 816, then the system will be the one shown in **Figure 5**, where conferencing application 401 and conference component 400 are not executing. If conferencing application 401 is not listening in the same way as it was when conferencing application 401 set-up call director 502 to listen for incoming calls, then the system will be as shown in **Figure 6**, where conferencing application 401 and conference component 400 are executing even though they are not processing any incoming calls.

While the present invention has been particularly described with reference to the various figures, it should be understood that the figures are for illustration only and should not be taken as limiting the scope of the invention. Many changes and modifications may be made to the invention, by one having ordinary skill in the art, without departing from the spirit and scope of the invention.

CLAIMS

What is claimed is:

1. In a computer system having a memory, a processor, and a network interface, a method comprising the steps of:

receiving an incoming call signal on said network interface;

processing said incoming call signal to detect an intended recipient application; and

launching said intended recipient application.

2. The method of claim 1, wherein said step of processing said incoming call signal comprises the steps of:

parsing said incoming call signal to determine a signal type; and a signal port; and

determining said intended recipient application based on said signal type and said signal port.

3. The method of claim 1, wherein said step of launching said intended recipient application comprises the steps of:

determining said intended recipient application based on said signal type and said signal port;

locating said intended recipient application using an application signature; and

signaling a process manager to launch said intended recipient application.

1 4. The method of claim 1, further comprising the steps of:
2 loading a call processing module into said memory; and
3 initializing said call processing module to process
4 calls using said network interface.

1 5. The method of claim 2, wherein said step of loading said call
2 processing module into said memory comprises the steps of:
3 loading a call directing component;
4 loading a first conference component;
5 loading a first transport component; and
6 loading a first network component.

1 6. The method of claim 5, wherein said step of initializing said
2 call processing module comprises the steps of:
3 initializing said first network component to operate
4 with said network interface;
5 initializing said call directing component to monitor
6 for said incoming call signal;
7 initializing said first transport component to receive
8 said incoming call signal; and
9 initializing said first conference component to transfer
10 said incoming call signal.

1 7. The method of claim 1, further comprising the steps of:
2 receiving an initialization message from said intended
3 recipient application; and

4 removing said intended recipient application from an
5 internal list if said initialization message does not correspond
6 to an expected message.

1 8. In a computer system having a memory, a processor, and a
2 network interface, an apparatus comprising:
3 a call directing module;
4 a process manager coupled to said call directing module;
5 and,
6 a conferencing component coupled to said network
7 interface; and said call directing module
8 said conferencing component containing a circuit for
9 notifying said call directing module upon receipt of an incoming
10 call and causing said call director to signal said process manager
11 to activate a conferencing application.

1 9. An apparatus comprising:
2 a processor;
3 a memory coupled to said processor;
4 a network interface coupled to said processor;
5 said memory configured to cause said processor to:
6 receiving an incoming call signal on said network
7 interface;
8 processing said incoming call signal to detect an
9 intended recipient application; and

10 launching a conferencing application if said
11 intended recipient application is said conferencing
12 application.



ABSTRACT OF THE INVENTION

1 in a computer system having a memory, a processor, and a
2 network interface, a method for dynamically launching a
3 conferencing application upon the receipt of an incoming call
4 having the steps of: receiving an incoming call signal on the
5 network interface; processing the incoming call signal to detect
6 an intended recipient application; and launching the intended
7 recipient application.

8 An apparatus for dynamically launching a conferencing
9 application upon the receipt of an incoming call having a call
10 directing module; a process manager coupled to the call directing
11 module; and, a conferencing component coupled to the network
12 interface; and the call directing module; the conferencing
13 component containing a circuit for notifying the call directing
14 module upon receipt of an incoming call and causing the call
15 director to signal the process manager to activate a conferencing
16 application.



DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below, next to my name.

I believe I am the original, first, and sole inventor of the subject matter which is claimed and for which a patent is sought on the invention entitled

**METHOD AND APPARATUS FOR DYNAMIC LAUNCHING OF A
TELECONFERENCING APPLICATION UPON RECEIPT OF A CALL**

the specification of which

XXXX

is attached hereto.
was filed on May 8, 1996 as
United States Application Number 08/646,503
or PCT International Application Number _____
and was amended on _____
(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claim(s), as amended by any amendment referred to above. I do not know and do not believe that the claimed invention was ever known or used in the United States of America before my invention thereof, or patented or described in any printed publication in any country before my invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, and that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months (for a utility patent application) or six months (for a design patent application) prior to this application.

I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d), of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

<u>Prior Foreign Application(s)</u>			<u>Priority Claimed</u>	
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
_____	_____	_____	_____	_____
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
_____	_____	_____	_____	_____
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
_____	_____	_____	_____	_____

I hereby claim the benefit under title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below

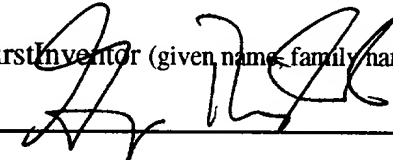
(Application Number)	Filing Date
(Application Number)	Filing Date

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Number)	Filing Date	(Status -- patented, pending, abandoned)
(Application Number)	Filing Date	(Status -- patented, pending, abandoned)

I hereby appoint Aloysius T. C. AuYeung, Reg. No. 35,432; William Thomas Babbitt, Reg. No. P39,591; Bradley J. Bereznak, Reg. No. 33,474; Michael A. Bernadicou, Reg. No. 35,934; Roger W. Blakely, Jr., Reg. No. 25,831; Thomas M. Coester, Reg. No. P39,637; William Donald Davis, Reg. No. 38,428; Daniel M. De Vos, Reg. 37,813; Karen L. Feisthamel, Reg. No. P40,264; Scot A. Griffin, Reg. No. 38,167; David R. Halvorson, Reg. No. 33,395; Brian Don Hickman, Reg. No. 35,894; Eric Ho, Reg. No. P39,711; George W Hoover II, Reg. No. 32,992; Eric S. Hyman, Reg. No. 30,139; Jeff D. Jacobs, Reg. No. P40,029; Dag H. Johansen, Reg. No. 36,172; Stephen L. King, Reg. No. 19,180; Daniel C. Mallery, Reg. 33,532; Michael J. Mallie, Reg. No. 36,591; Kimberley G. Nobles, Reg. No. 38,255; Ronald W. Reagin, Reg. No. 20,340; James H. Salter, Reg. No. 35,668; William W. Schaal, Reg. No. P39,018; James C. Scheller, Reg. No. 31,195; Edward W. Scott, IV, Reg. No. 36,000; Maria McCormack Sobrino, Reg. No. 31,639; Stanley W. Sokoloff, Reg. No. 25,128; Allan T. Sponseller, Reg. No. 38,318; Steven R. Sponseller, Reg. No. 39,384; David R. Stevens, Reg. No. 38,626; Edwin H. Taylor, Reg. No. 25,129; Lester J. Vincent, Reg. No. 31,460; John Patrick Ward, Reg. No. P40,216; Ben J. Yorks, Reg. No. 33,609; and Norman Zafman, Reg. No. 26,250; my attorneys; and Roland B. Cortes, Reg. No. 39,152; Gary B. Goates, Reg. No. 35,159; Thomas X. Li, Reg. No. 37,079; and Edwin A. Sloane, Reg. No. 34,728; my patent agents; of BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, with offices located at 12400 Wilshire Boulevard, 7th Floor, Los Angeles, California 90025, telephone (310) 207-3800, with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith. I also hereby appoint Mark Aaker, Reg. No. 32,667, Paul D. Carmichael, Reg. No. 18,679; Vernon Randall Gard, Reg. No. 33,886; David J. Larwood, Reg. No. 33,191; Richard Liu, Reg. No. 34,377; Helene Plotka Workman, Reg. No. 35,981; Edward W. Scott, IV, Reg. No. 36,000; and Nancy R. Simon, Reg. No. 36,930; my attorneys; of APPLE COMPUTER, INC., located at 1 Infinite Loop, MS: 38-PAT, Cupertino, California 95014, telephone (408)974-9453, will full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole/First Inventor (given name, family name) Guy Riddle
Inventor's Signature  Date 26 June 1996
Residence Los Gatos, California Citizenship U.S.A.
(City, State) (Country)
Post Office Address 18243 Knute Road
Los Gatos, California 95030

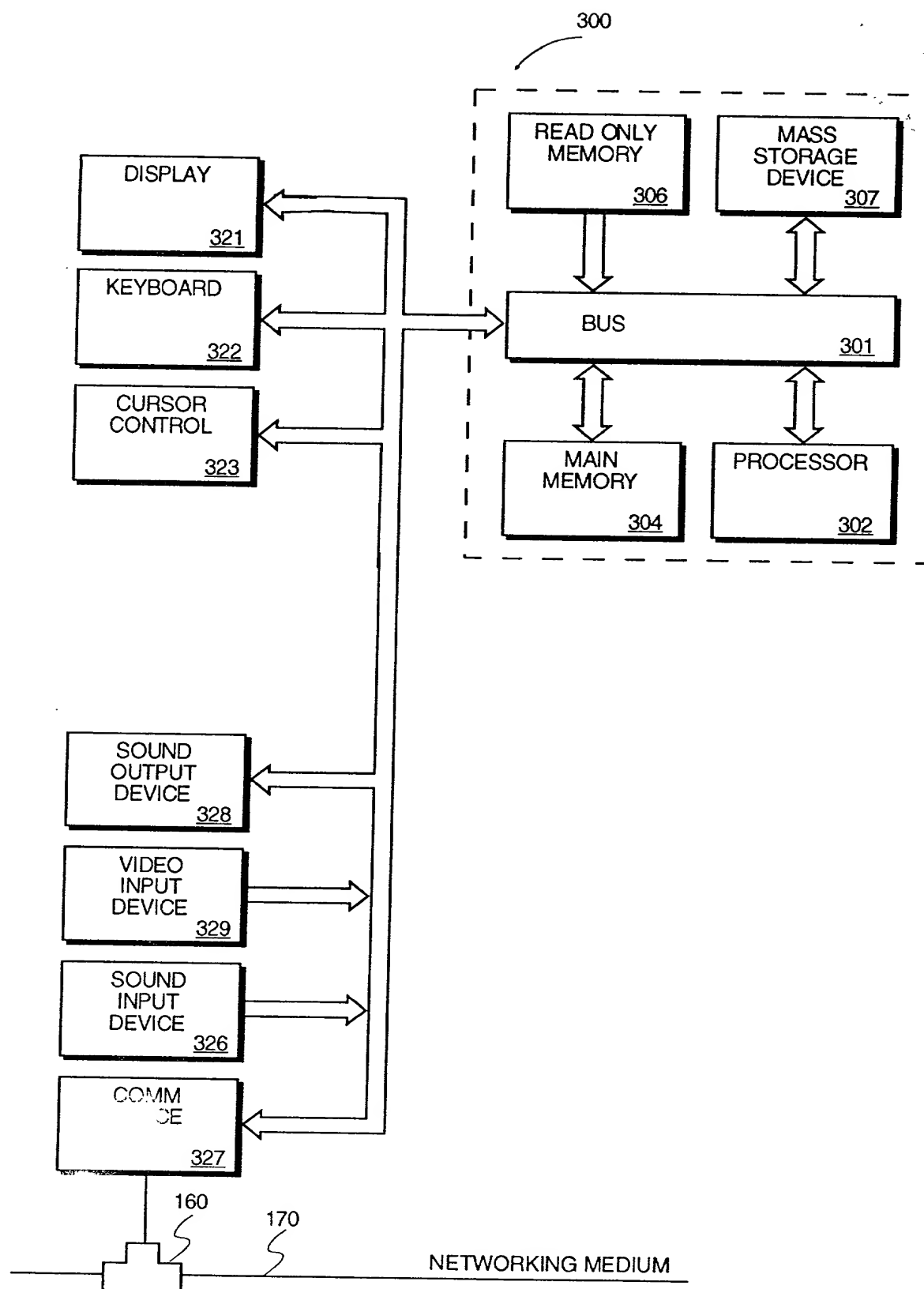
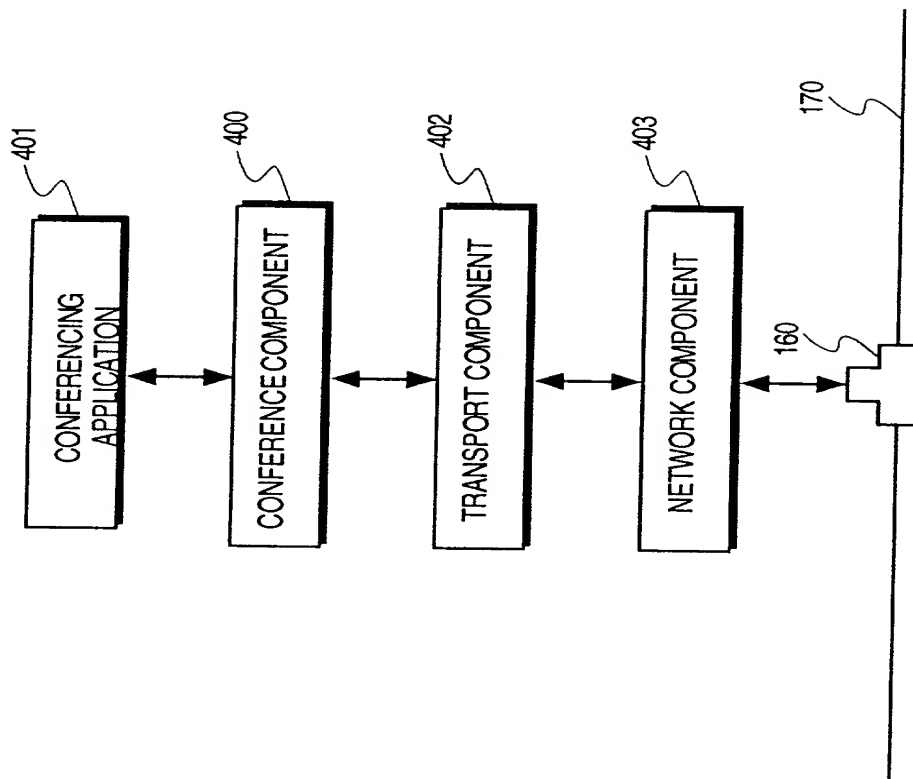
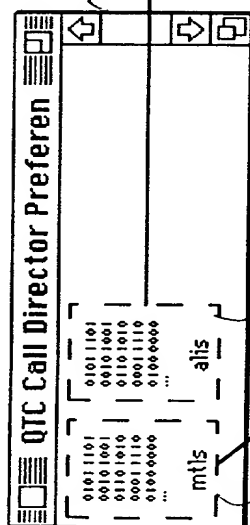
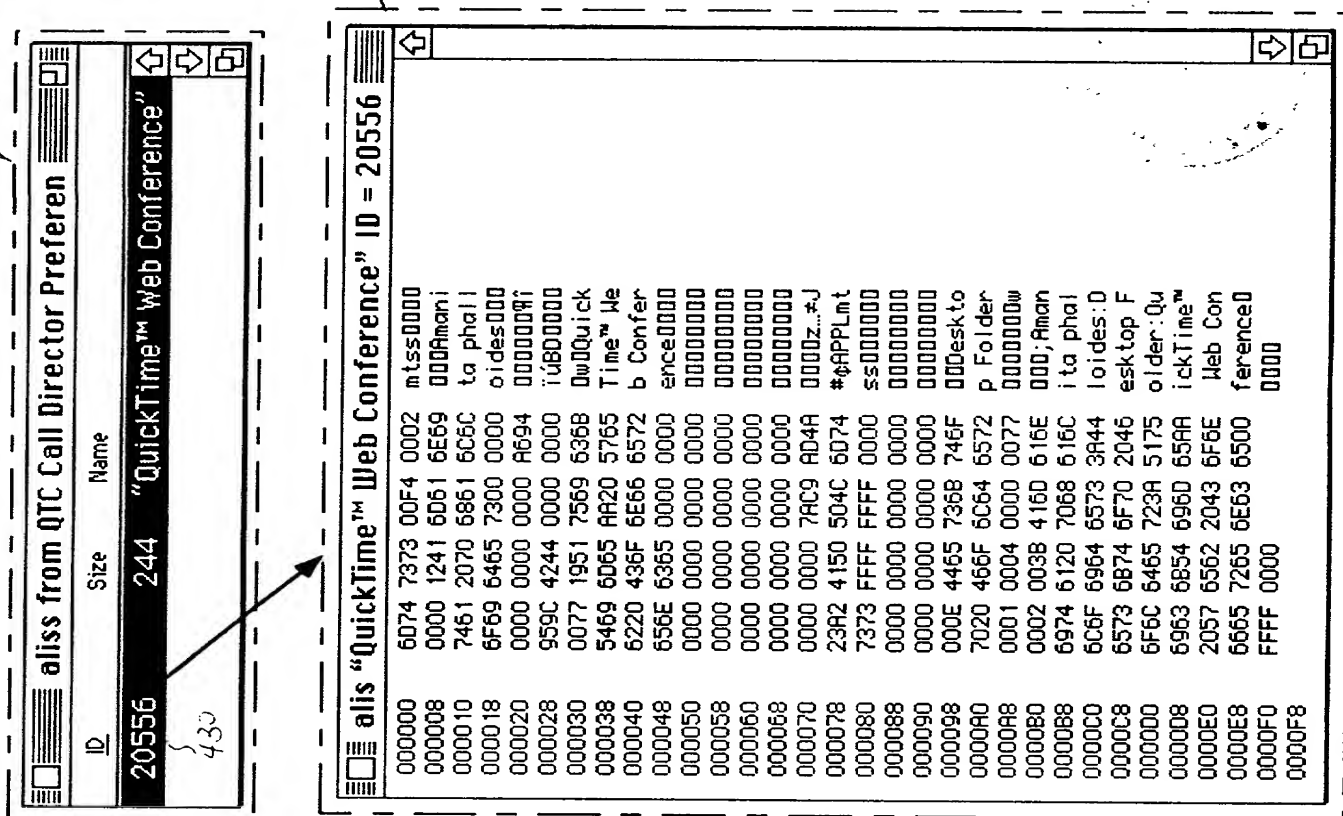
*Figure 2*

Figure 3

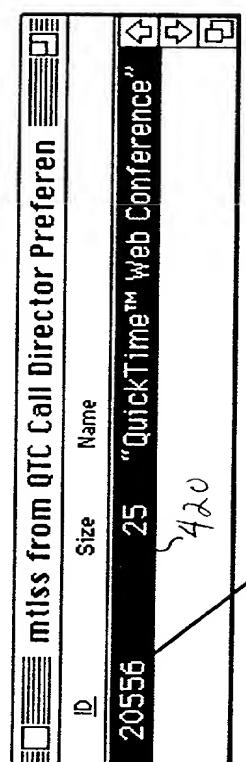


418

432



416



422

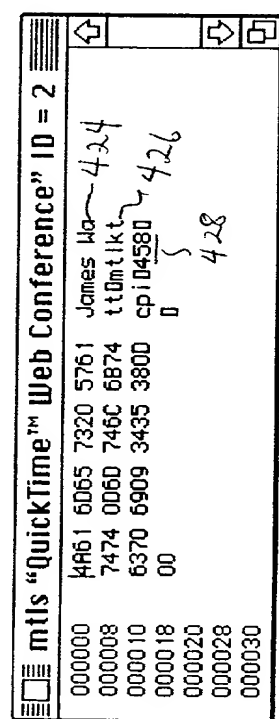
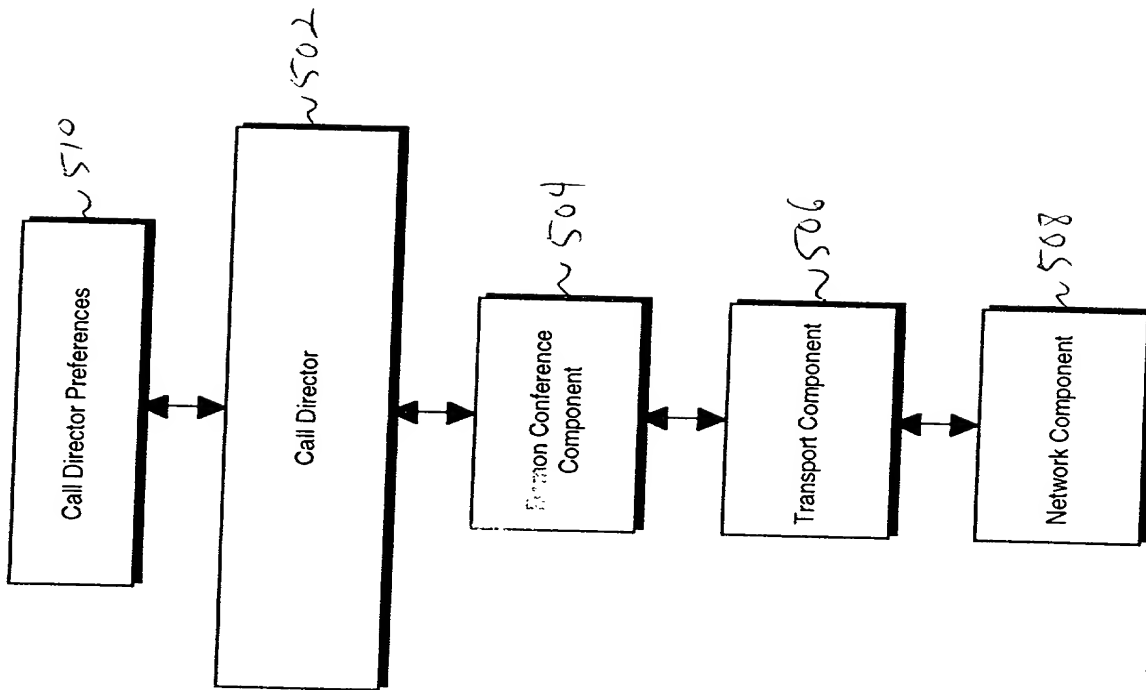


Figure 4

*Figure 5*

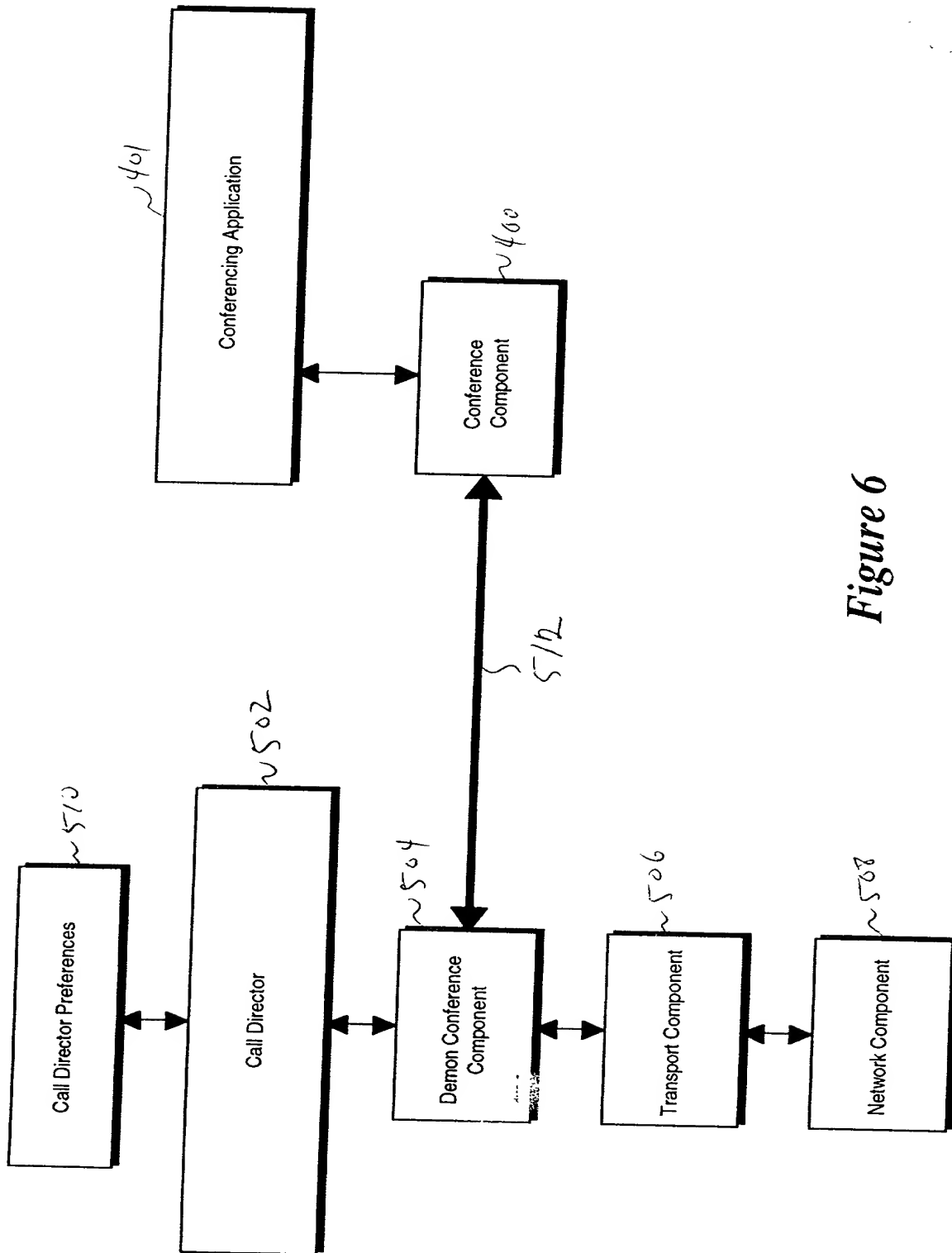


Figure 6

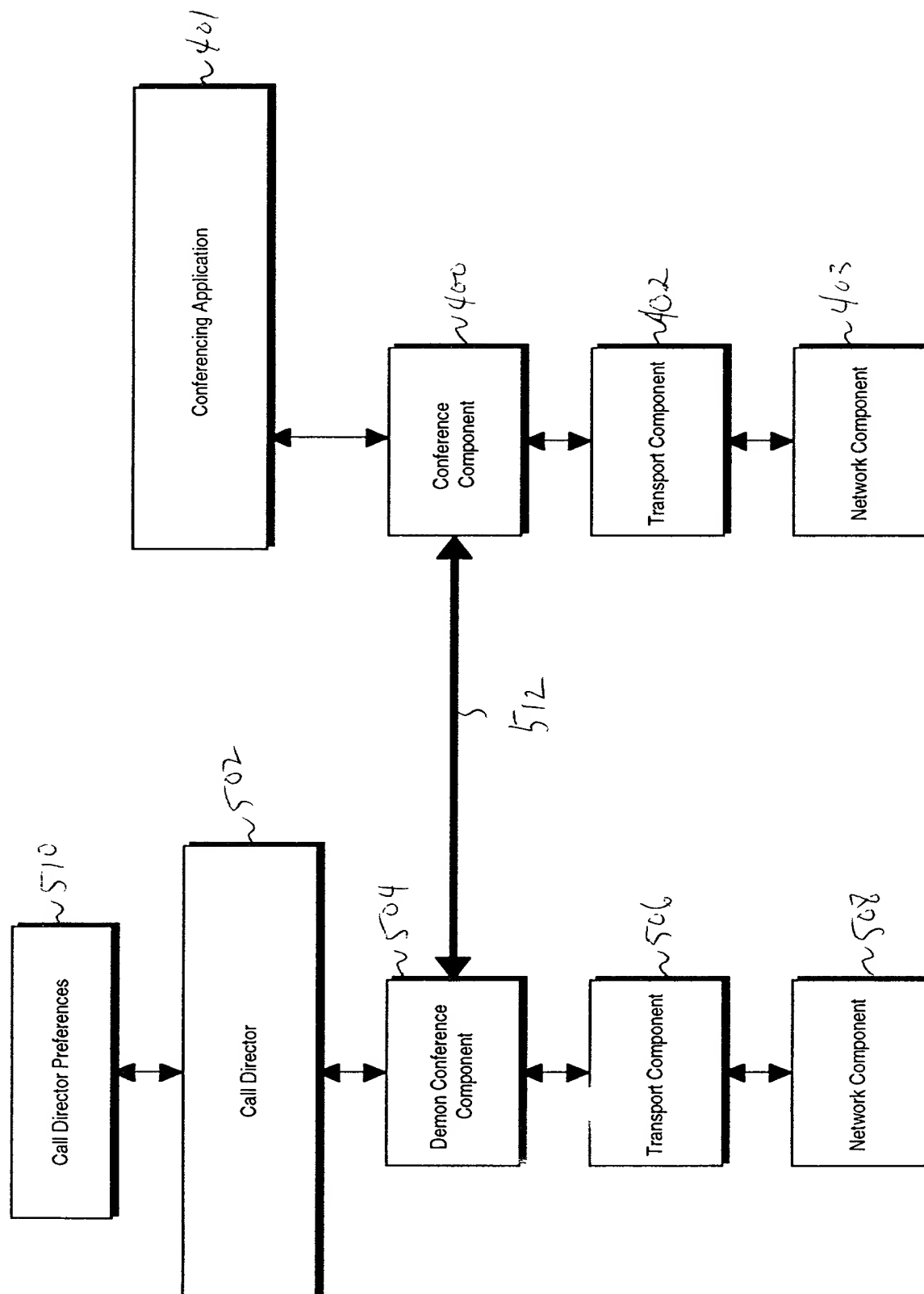


Figure 7

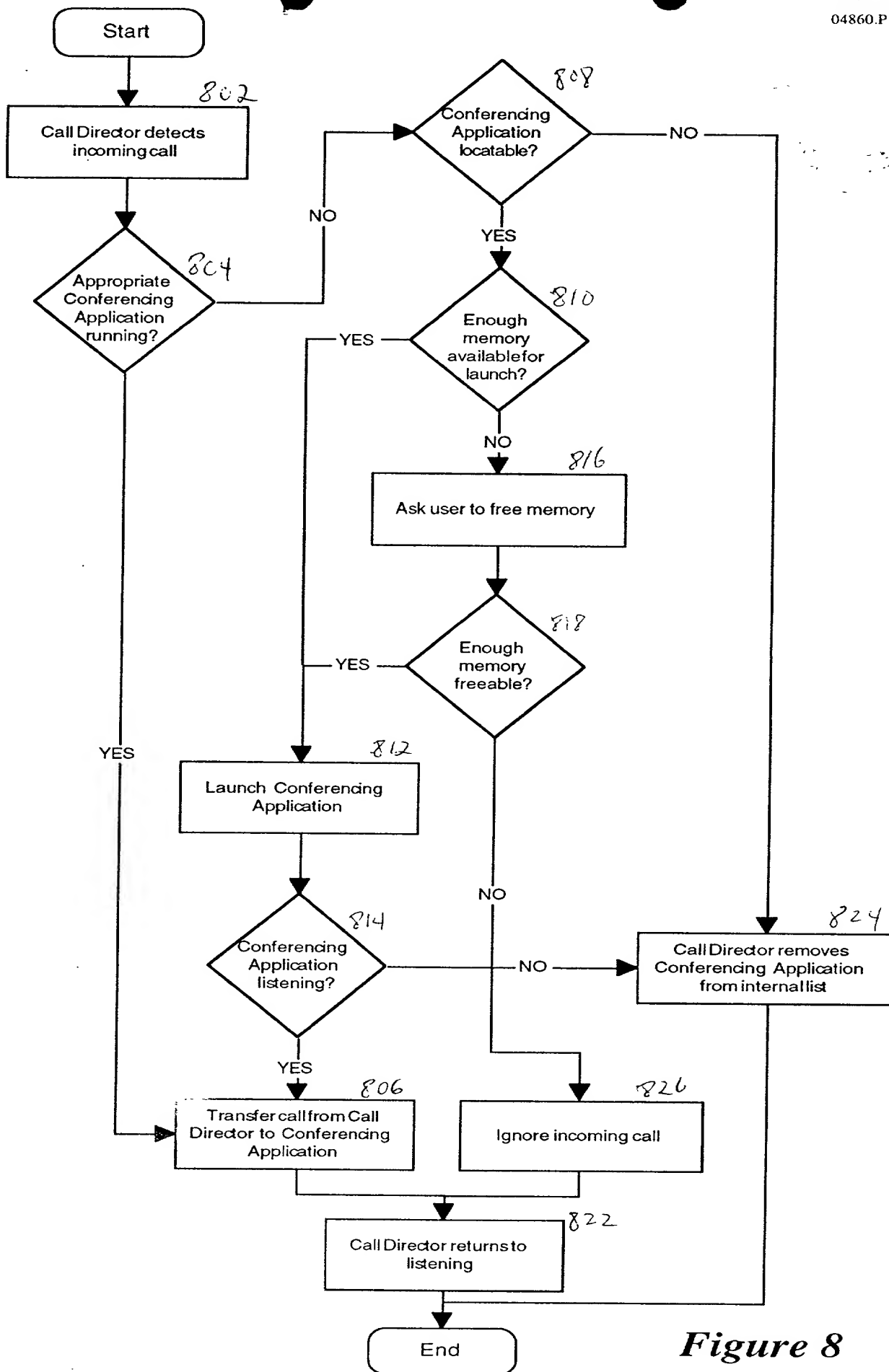


Figure 8

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL

Note: Effective October 1, 1997

Patent fees are subject to annual revision

MAR 30 1998

Complete If Known

Application Number	08/646,503
Filing Date	May 8, 1996
First Named Inventor	Guy Riddle
Group Art Unit	Unknown
Examiner Name	Not assigned
Attorney Docket No.	004860.P1937

TOTAL AMOUNT OF PAYMENT

(\$170.00)

METHOD OF PAYMENT (check one)**FEE CALCULATION (continued)**

1. ☒ The Commissioner is hereby authorized to charge indicated fees & credit any overpayments to:

Acct # 02-2666

Acct Name Blakely Sokoloff Taylor & Zafman

- ☒ Charge any add'l fee required under 37 CFR 1.16 & 1.17 ☐ Charge issue fee set in 37 CFR 1.18 at the mailing of the Notice of Allowance

2. ☒ Payment Enclosed:

☒ Check ☐ Money Order ☐ Other
Fee Calculation**I. Filing Fee**

Large Entity Code	Large Entity Fee (\$)	Small Entity Code	Small Entity Fee (\$)	Fee Description	Fee Paid
101	790	201	395	Utility filing fee	
106	330	206	165	Design filing fee	
107	540	207	270	Plant filing fee	
108	790	208	395	Reissue filing fee	
114	150	214	75	Provisional filing fee	
Subtotal (1)					(\$)

2. Claims

	*	Extra	Fee from Below	Fee Paid
Total Claims	20 -	<input type="checkbox"/>	X	
Ind. Claims	3 -	<input type="checkbox"/>	X	
Multiple Dependent Claims				

Large Entity Code	Large Entity Fee (\$)	Small Entity Code	Small Entity Fee (\$)	Fee Description	Fee Paid
103	22	203	11	Claims in excess of 20	
102	82	202	41	Independent claims in excess of 3	
104	270	204	135	Multiple dependent claim	
109	82	209	41	Reissue independent claims over original patent	
110	22	210	11	Reissue claims in excess of 20 & over original patent	
Subtotal (2)					(\$)

3 Additional Fees

Large Entity Code	Large Entity Fee (\$)	Small Entity Code	Small Entity Fee (\$)	Fee Description	Fee Paid
105	130	205	65	Surcharge-late filing fee or oath	\$130
127	50	227	25	Surcharge-late provisional filing fee or cover sheet	
139	130	139	130	Non-English specification	
147	2520	147	2520	For filing a request for reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1840*	113	1840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within 1st month	
116	400	216	200	Extension for reply within 2nd month	
117	950	217	475	Extension for reply within 3rd month	
118	1510	218	755	Extension for reply within 4th month	
128	2060	228	1030	Extension for reply within 5th month	
119	310	219	155	Notice of Appeal	
120	310	220	155	Filing a brief in support of appeal	
121	270	221	135	Request for oral hearing	
138	1510	138	1510	Petition to institute a public use proceeding	
140	110	240	55	Petition to Revive-unavoidable	
141	1320	241	660	Petition to Revive-unintentional	
142	1320	242	660	Utility issue fee (or reissue)	
143	450	243	225	Design issue fee	
144	670	244	335	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Petitions related to provisional applications	
126	240	126	240	Submission of IDS	
581	40	581	40	Recording each patent assignment per property	\$40
146	790	246	395	Filing a submission after final rejection	
149	790	249	395	For each add'l invention to be examined	

Other fee (specify) Postcard

Other fee (specify)

*Reduced by Basic Filing Fee Paid

Subtotal (3) (\$170.00)

SUBMITTED BY BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

COMPLETE (if applicable)

Name Eric S. Hyman

Reg. Number 30,139

Signature

Date 3/27/98

Deposit Acct User ID 02-2666

*Highest number of claims previously paid for in an amendment is being transmitted.

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

FEE TRANSMITTALNote: Effective October 1, 1997
Patent fees are subject to annual revision

MAR 30 1998

Complete If Known

Application Number	08/646,503
Filing Date	May 8, 1996
First Named Inventor	Guy Riddle
Group Art Unit	Unknown
Examiner Name	Not assigned
Attorney Docket No.	004860.P1937

TOTAL AMOUNT OF PAYMENT

(\$170.00)

METHOD OF PAYMENT (check one)

- 1.
- ☒
- The Commissioner is hereby authorized to charge indicated fees & credit any overpayments to:

Acct # 02-2666

Acct Name Blakely Sokoloff Taylor & Zafman

- ☒
- Charge any add'l fee required under 37 CFR 1.16 & 1.17
- ☐
- Charge issue fee set in 37 CFR 1.18 at the mailing of the Notice of Allowance

- 2.
- ☒
- Payment Enclosed:

☒ Check
 ☐ Money Order
 ☐ Other
Fee Calculation**1. Filing Fee**

Large Entity Code	Large Entity Fee (\$)	Small Entity Code	Small Entity Fee (\$)	Fee Description	Fee Paid
101	790	201	395	Utility filing fee	
106	330	206	165	Design filing fee	
107	540	207	270	Plant filing fee	
108	790	208	395	Reissue filing fee	
114	150	214	75	Provisional filing fee	

Subtotal (1) (\$)

2. Claims

	*	Extra	Fee from Below	Fee Paid
Total Claims	20 -		X	
Ind. Claims	3 -		X	
Multiple Dependent Claims				

Large Entity Code	Large Entity Fee (\$)	Small Entity Code	Small Entity Fee (\$)	Fee Description
103	22	203	11	Claims in excess of 20
102	82	202	41	Independent claims in excess of 3
104	270	204	135	Multiple dependent claim
109	82	209	41	Reissue independent claims over original patent
110	22	210	11	Reissue claims in excess of 20 & over original patent

Subtotal (\$)
(2)**FEE CALCULATION (continued)****3 Additional Fees**

Large Entity Code	Large Entity Fee (\$)	Small Entity Code	Small Entity Fee (\$)	Fee Description	Fee Paid
105	130	205	65	Surcharge-late filing fee or oath	\$130
127	50	227	25	Surcharge-late provisional filing fee or cover sheet	
139	130	139	130	Non-English specification	
147	2520	147	2520	For filing a request for reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1840*	113	1840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within 1st month	
116	400	216	200	Extension for reply within 2nd month	
117	950	217	475	Extension for reply within 3rd month	
118	1510	218	755	Extension for reply within 4th month	
128	2060	228	1030	Extension for reply within 5th month	
119	310	219	155	Notice of Appeal	
120	310	220	155	Filing a brief in support of appeal	
121	270	221	135	Request for oral hearing	
138	1510	138	1510	Petition to institute a public use proceeding	
140	110	240	55	Petition to Revive-unavoidable	
141	1320	241	660	Petition to Revive-unintentional	
142	1320	242	660	Utility issue fee (or reissue)	
143	450	243	225	Design issue fee	
144	670	244	335	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Petitions related to provisional applications	
126	240	126	240	Submission of IDS	
581	40	581	40	Recording each patent assignment per property	\$40
146	790	246	395	Filing a submission after final rejection	
149	790	249	395	For each add'l invention to be examined	

Other fee (specify) Postcard

Other fee (specify)

*Reduced by Basic Filing Fee Paid

Subtotal (3) (\$170.00)

SUBMITTED BY BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

COMPLETE (if applicable)

Name Eric S. Hyman

Reg. Number 30,139

Signature

Date

3/27/98

Deposit Acct User ID 02-2666

*Highest number of claims previously paid for in an amendment is being transmitted.

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.



Our Ref. No.: 004860.P1937

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Guy Riddle

Serial No.: 08/646,503

Filed: May 8, 1996

For: METHOD AND APPARATUS FOR
DYNAMIC LAUNCHING OF A
TELECONFERENCING
APPLICATION UPON RECEIPT OF
A CALL

Examiner: Not Assigned

Art Group: Unknown

RESPONSE TO NOTICE TO FILE MISSING PARTS OF APPLICATION

BOX MISSING PARTS

Assistant Commissioner For Patents
Washington, D.C. 20231

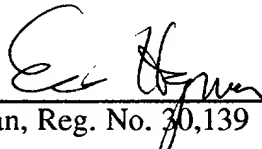
Sir:

In response to the communication from the Patent Office dated March 4, 1998, on the above-referenced utility patent application, enclosed herewith are Form PTO-1533, the executed Declaration and Power of Attorney, Assignment and Assignment Cover Sheet, as well as our check numbered 8231 in the amount of one hundred thirty dollars (\$130.00) as the total amount owed for a large entity, and check number 8230 in the amount of forty dollars (\$40.00) for Recordation of the Assignment. Please record and return the recorded assignment. Also please charge any additional fees to Deposit Account No. 02-2666.

Respectfully submitted,

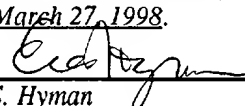
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: March 27, 1998


Eric S. Hyman, Reg. No. 30,139

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231 on: March 27, 1998.


Eric S. Hyman

3/27/98
Date

12400 Wilshire Boulevard, Seventh Floor
Los Angeles, California 90025
(310) 207-3800